

CPSC 217: Introduction to Computer Science for Multidisciplinary Studies I

Assignment 1: First Program/Data and Information

Weight: 6%

Collaboration

Discussing the assignment requirements with others is a reasonable thing to do, and an excellent way to learn. However, the work you hand-in must ultimately be your work. This is essential for you to benefit from the learning experience, and for the instructors and TAs to grade you fairly. Handing in work that is not your original work, but is represented as such, is plagiarism and academic misconduct. Penalties for academic misconduct are outlined in the university calendar.

Here are some tips to avoid plagiarism in your programming assignments.

1. Cite all sources of code that you hand-in that are not your original work. You can put the citation into comments in your program. For example, if you find and use code found on a web site, include a comment that says, for example:

```
# the following code is from  
https://www.quackit.com/python/tutorial/python\_hello\_world.cfm.
```

Use the complete URL so that the marker can check the source.

2. Citing sources avoids accusations of plagiarism and penalties for academic misconduct. **However, you may still get a low grade if you submit code that is not primarily developed by yourself. Cited material should never be used to complete core assignment specifications. You can and should verify and code you are concerned with your instructor/TA before submit.**
3. Discuss and share ideas with other programmers as much as you like, but make sure that when you write your code that it is your own. A good rule of thumb is to wait 20 minutes after talking with somebody before writing your code. If you exchange code with another student, write code while discussing it with a fellow student, or copy code from another person's screen, then this code is not yours.
4. **Collaborative coding is strictly prohibited. Your assignment submission must be strictly your code.** Discussing anything beyond assignment requirements and ideas is a strictly forbidden form of collaboration. This includes sharing code, discussing code itself, or modelling code after another student's algorithm. **You can not use (even with citation) another student's code.**
5. Making your code available, even passively, for others to copy, or potentially copy, is also plagiarism.
6. We will be looking for plagiarism in all code submissions, possibly using automated software designed for the task. For example, see Measures of Software Similarity (MOSS - <https://theory.stanford.edu/~aiken/moss/>).
7. Remember, if you are having trouble with an assignment, it is always better to go to your TA and/or instructor to get help than it is to plagiarize. A common penalty is an F on a plagiarized assignment.

Late Penalty

Late assignments will not be accepted.

Goal

Writing a first program in Python and data/information questions.

Technology

Python 3, SimpleGraphics.py

Submission Instructions

This assignment includes two parts. The first part requires you to write a computer program using **Python**. The second part of this assignment consists of answered questions. Both parts must be submitted electronically. Use the **Assignment 1** drop box in D2L for the electronic submission. You can submit multiple times over the top of a previous submission. Do not wait until the last minute to attempt to submit. You are responsible if you attempt this, and time runs out. Your assignment must be completed in **Python 3** and be executable with Python version **3.11.8+**. For graphical drawing you must use the **SimpleGraphics.py** library. **Do not import any additional libraries to complete assignment without instructor approval.**

Posting Submissions for Public Viewing:

In previous years, we have received many creative and highly artistic submissions for the first part of this assignment. As such, we plan to post the images that are created on the course webpage so that others can view them. Your image will be posted anonymously, unless you choose to include your name as part of the image that you create. Please do not put your student number on your image. If you do not wish to have your image included on the course website, then please send an email to jwhudson@ucalgary.ca clearly stating such.

Description

Part 1: Creating a Graphical Python Program

In this part of the assignment, you will create a small graphical **Python** program. The program will **take four integers as input** from the user. **Its output will be a landscape with certain features. A rocket and a moon in this artistic image will be centered on the positions specified by the user.**

*For example, if the user enters 400 and 300 as first two inputs then the rocket should be centered in 800 by 600 window. If the user then enters 0 and 0, then the moon should be centered in the upper left corner of the window, with part of it cut off by the edge of the window. Use the **SimpleGraphics** library that you were introduced to in tutorials to complete this task.*

Getting Input:

Read the input using the techniques that we have discussed in class. **Display a prompt in the terminal window** with a print statement, or by providing a parameter to the `input` function. Note that the user will enter their input in the terminal window, not in the graphics window.

Recommended Algorithm:

There are many ways to create the program described in this section of the assignment. If you are having trouble getting started, you may want to consider creating your solution by using the following steps:

1. Import the **SimpleGraphics** library
2. Draw the landscape consisting of grass, sky, and at least one other feature (cloud and/or tree and/or mountain and/or etc.).
3. Prompt the user to enter the **x-position** by displaying a message in the terminal
4. Read the **x-position** from the user as a number
5. Prompt the user to enter the **y-position** by displaying a message in the terminal
6. Read the **y-position** from the user as a number
7. Draw the **rocket** by calling functions in the **SimpleGraphics** library. These function calls will likely include calculations involving **x** and **y** (*attend tutorial for introduction to how to do these calculations*)
8. Prompt the user to enter a second **x-position** by displaying a message in the terminal
9. Read the **x-position** from the user as a number
10. Prompt the user to enter a second **y-position** by displaying a message in the terminal
11. Read the **y-position** from the user as a number
12. Draw the **moon** by calling functions in the **SimpleGraphics** library. These function calls will likely include calculations involving **x** and **y** (*attend tutorial for introduction to how to do these calculations*)

Additional Specifications:

Ensure that your program meets all the following requirements as these are part of your grade:

- The image generated by your program should be a **landscape (grass, sky, other features) with some sort of a movable rocket** that uses a good portion the drawing area, and a **movable moon**.
- The **rocket&moon** generated by your program must use a total of at least **4 distinct colors**.
- The **rocket&moon** generated by your program must use a total of at least **4 different graphics primitives** such as ellipses, polygons, rectangles, text, etc.
- Your program **may not load images** or use other features that **require us to have files beyond your .py file**. *Please use standard fonts such as Arial or Times if you choose to include text in your image.*

- Do not resize the window.
- **Your program should read only four input values** – the x-position and the y-position of the center of the **house** and **sun**.
- Your program must include **appropriate comments, including a comment at the top of your file** which includes your name and student number and describes the purpose of your program. **There should also be comments within the program that indicate which lines of code draw different parts of the house/sun/landscape** (roof, walls, doors, windows, etc.).
- This program is unique for the semester regarding removing magic numbers. You **do not** need to replace magic numbers used to control the positions of shapes with named constants.
- **Do not display your student number as part of the image.**

Part 2: Information and Data

Solve the following problems and submit your answers as a document called CPSC217S24A1-Name.pdf (Ex. CPSC217S24A1-Hudson). An electronic submission is required for this part of the assignment. For base conversions **you must still show your work and the base of your answers must be clear. Grading will primarily be based on the answer, but full credit is only given if your work to get that answer is included in your submission. [base conversion work can be written and scanned, or typed electronically]**

- [5 marks] Convert the following base 10 numbers to binary:
 - 758
 - 512
 - 4336
 - 2246
 - 899263
- [5 marks] Repeat question 1, converting each base 10 value to base 9:
- [5 marks] Repeat question 1, converting each base 10 value to hexadecimal:
- [5 marks] Repeat question 1, converting each base 10 value to base 6:
- [5 marks] Convert the following base 2 numbers to decimal:
 - 100
 - 1 1011
 - 100 0110
 - 1 0111 1000
 - 101 1100 1010 0010 0110
- [5 marks] Convert the following numbers:
 - 2401 base 5 to base 10
 - 2351 base 6 to base 10
 - 1c3 base 16 to base 10
 - 170 base 8 to base 10

- e. 7261 base 9 to base 10
7. [8 marks] Convert the following numbers:
- a. E1 base 16 to base 7
 - b. 101 0101 0100 base 2 to base 4
 - c. DAC base 15 to base 20
 - d. 3041 base 6 to base 16
8. [12 marks] Answer the following questions. Response to each question should be brief (3 sentences or less).
- a. In ASCII, what value represents the letter 'J'?
 - b. Is the value assigned to the letter 'J' arbitrary, or was it selected for a good reason? Justify your answer.
 - c. In ASCII, what value represents the letter 'h'?
 - d. Is the value assigned to the letter 'H' arbitrary, or was it selected for a good reason? Justify your answer.
 - e. In ASCII, what value represents the character '7'?
 - f. Is the value assigned to the letter '7' arbitrary? Justify your answer.
 - g. In ASCII, what value represents the character '&'?
 - h. Is the value assigned to the character '&' arbitrary? Justify your answer.
 - i. What is UTF-8?
 - j. What advantages does UTF-8 have compared to ASCII?
 - k. What disadvantages does UTF-8 have compared to ASCII?
 - l. Why are floating point numbers only an approximation of real numbers?

Submit the following using the Assignment 1 Dropbox in D2L:

- 1. CPSC217S24A1-Name.py
- 2. CPSC217S24A1-Name.pdf

Grading:

Part one of the assignment will be graded out of 50, with the grade based on the program's level of functionality and conformance to the specifications. A small number of bonus marks may be awarded to particularly impressive submissions. Part two of the assignment will be graded based on the number of correct answers provided.

The total mark achieved for the assignment will be translated to a letter grade using the following table:

Mark	Letter
100+	A+
95 to 99	A
90 to 94	A-
87 to 89	B+

83 to 86	B
80 to 82	B-
77 to 79	C+
73 to 76	C
70 to 72	C-
65 to 69	D+
60 to 64	D
0 to 59	F

As a reminder, the University of Calgary assigns the following meaning to letter grades:

A: Excellent – Superior performance showing a comprehensive understanding of the subject matter

B: Good – Clearly above average performance with generally complete knowledge of the subject matter

C: Satisfactory – Basic understanding of the subject matter

D: Minimal Pass – Marginal performance; Generally insufficient preparation for subsequent courses in the same subject

F: Fail – Unsatisfactory performance